

Thesis for the Degree of Licentiate of Engineering

---

# On Clock Synchronization in Wireless Networks Using Parameter Estimation and Consensus Techniques

Wanlu Sun



**CHALMERS**

Communication Systems Group  
Department of Signals and Systems  
Chalmers University of Technology

Gothenburg, Sweden 2013

Sun, Wanlu

On Clock Synchronization in Wireless Networks Using Parameter Estimation and Consensus Techniques.

Department of Signals and Systems  
Technical Report No. R022/2013  
ISSN 1403-266X

Communication Systems Group  
Department of Signals and Systems  
Chalmers University of Technology  
SE-412 96 Göteborg, Sweden  
Telephone: + 46 (0)31-772 1000  
Email: wanlu@chalmers.se

Copyright ©2013 Wanlu Sun  
except where otherwise stated.  
All rights reserved.

This thesis has been prepared using L<sup>A</sup>T<sub>E</sub>X.

Printed by Chalmers Reproservice,  
Göteborg, Sweden, November 2013.

To my parents and Wei

*“Time is God’s way of keeping everything from happening at once.”*

– Unknown



# Abstract

Time agreement in a network is a critical issue for communications. While clock synchronization in wired networks is a well studied problem, the wireless medium presents extra challenges. For example, in wireless sensor networks which may contain several thousands of cheap and small sensors, the scalability and energy conservation are two major considerations in the design of synchronization. Moreover, in wireless ad hoc networks which have no infrastructure or limited infrastructure support, self-organization is an important concern. Besides, in vehicular networks, high mobility will require synchronization to be robust against node failures and dynamic topologies.

Clock synchronization in wireless networks can be divided into pairwise and network-wide clock synchronization. Pairwise clock synchronization is designed for two nodes, where the goal is to synchronize one node to the other one by transferring timing messages. In this case, the synchronization problem can be transformed into a parameter estimation problem, since the random delays during the transmission lead to inaccuracy of timing messages. On the other hand, the network-wide clock synchronization is designed for synchronizing multiple nodes in a network, which is inherently a consensus problem. In this formulation, the main considerations are convergence speed and resilience to random delays.

In [Paper A], we investigate the pairwise clock synchronization problem based on a two-way message exchange mechanism with exponentially distributed random delays. A novel synchronization scheme is proposed to estimate the clock skew and offset by directly utilizing mean square error as the metric to be optimized. This consideration results in performance improvements compared to the existing synchronization methods, especially for a small number of observations and large standard deviation of random delays.

Furthermore, [Paper B] studies the pairwise clock synchronization problem for different random delay distributions. In this paper, for a Gaussian distribution, we derive the Cramér-Rao lower bound for the joint estimation of clock offset and skew, and propose a linear-biased estimator to improve the accuracy. For an exponential distribution, we extend our work in [Paper A] by deriving the Chapman-Robbins bound. Moreover, a simple and robust estimation algorithm is proposed to handle arbitrary delay distributions, which can achieve a good tradeoff between accuracy and complexity.

In [Paper C], we study the network-wide clock synchronization problem. A distributed consensus synchronization algorithm is proposed when using a random access protocol for timing message broadcasts. In the absence of transmission delays, [Paper C] theoretically proves the convergence of the proposed scheme, which is further illustrated by the numerical results. On the other hand, when the transmission delays are also taken into account, the proposed approach still shows resilience.

**Keywords:** Wireless networks, clock synchronization, clock offset, clock skew, parameter estimation, distributed algorithms, consensus techniques, random delays.

# List of Included Publications

The thesis is based on the following appended papers:

- [A] W. Sun, F. Brännström, and E. G. Ström, “On clock offset and skew estimation with exponentially distributed delays,” in *Proceedings of IEEE International Conference on Communications (ICC)*, Budapest, Hungary, Jun. 2013.
  
- [B] W. Sun, E. G. Ström, and F. Brännström, “On clock offset and skew estimation with a two-way message exchange mechanism,” Submitted to *IEEE Transactions on Communications*, Nov. 2013.
  
- [C] W. Sun, E. G. Ström, F. Brännström, and M. R. Gholami, “Random broadcast based distributed consensus clock synchronization for mobile networks,” Submitted to *IEEE Transactions on Wireless Communications*, Nov. 2013.





# Acknowledgements

Well, even though there is a saying that you start your life after you finish your PhD, I am enjoying the life myself. Undoubtedly, that's because of all of you.

First and foremost, I would like to express my deepest gratitude to my main supervisor Prof. Erik G. Ström and my co-supervisor Prof. Fredrik Brännström. Your selfless guidance and education protect my research from going astray. Your unconditional support and encouragement let me go through difficult situations. Your endless patience and sincerity make me look into my mistakes in a soft way. Most importantly, you have corrected my attitude towards research and work, which I am sure I will benefit from in the whole life.

I would like to express my great appreciation to Prof. Shuguang (Robert) Cui and Dr. Ali Eslami at Texas A&M University for their creative inspirations, fruitful discussions, and warm hospitality. I believe our collaboration is not finished yet. Look forward to our future cooperations.

Many thanks to all the teachers who have given very interesting PhD courses. In particular, thanks to Prof. Lennart Svensson and Prof. Henk Wymeersch for introducing me filtering and Bayesian ideas. Additionally, thanks to Prof. Tommy Svensson for broadening my horizon regarding wireless communications during our interactions in the METIS project. Besides, I would like to thank all the administration staff at the S2 Department, with special thanks to Agneta, Lars, Natasha and Madeleine.

I would also like to thank the former and current members of the Communication Systems group. Dear friends, you really bring up the 'flavor' of my life. Thank you, Kasra, it is a nice experience to share the office with you. Thank you Yutao, for making my life much easier in Sweden. Thank you, Jingya, Mohammad, and Christopher, it is amazing that our technical discussions can be both serious and fun. Thank you, Naga, Christian and Srikar, for being in the small badminton group. We should continue it and make it regular. Thank you, Mikhail, Gabbo, Rajet, Reza, Tilak, Li, Erik, Rocco, Markus, Katharina, Keerthi, and Jessica, for all the wonderful parties we have had together. Besides, many thanks to all of my Chinese friends in Gothenburg for every excellent moment we have experienced together.

Finally, I would like to express my sincerest gratitude to my family. You are the sources of my happiness and courage. There is no word that is beautiful enough to describe my love. Dear mom and dad, thanks for the infinite love and support over the past 27 years. Dear Wei, thanks for your care, understanding, and being with me in all the good and bad times.

This work has been supported in part by SAFER-Vehicle and Traffic Safety Centre, Project A19 and by the Swedish Research Council project 2011-5824. Part of this work has

been performed in the framework of the FP7 project ICT-317669 METIS, which is partly funded by the EU. The research visit at Texas A&M University was partially supported by Ericsson's Research Foundation (Reference: FOSTIFT-13:043). The calculations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at C3SE.

Wanlu Sun  
Gothenburg, November 2013

# Acronyms

ADMM:	Alternating direction multiplier method
APP:	Application
CDF:	Cumulative distribution function
CHRB:	Chapman-Robbins bound
CRLB:	Cramér-Rao lower bound
GPS:	Global positioning systems
MAC:	Medium access control
ML:	Maximum likelihood
MVU:	Minimum variance unbiased
PDF:	Probability density function
PHY:	Physical
ppm:	Part per million
RMSE:	Root mean square error
SR:	Synchronization round
TDMA:	Time division multiple access
TSF:	Timing synchronization function
WSN:	Wireless sensor network



# Contents

<b>Abstract</b>	<b>i</b>
<b>List of Included Publications</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>Acronyms</b>	<b>vii</b>
 <b>I Overview</b>	 <b>1</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Scope and Aim of the Thesis . . . . .	2
1.3 Thesis Outline . . . . .	3
 <b>2 General Concepts on Clock Synchronization</b>	 <b>5</b>
2.1 Clock Models . . . . .	5
2.1.1 Hardware Clock Models . . . . .	5
2.1.2 Logical Clock Models . . . . .	7
2.2 Performance Metrics . . . . .	8
2.2.1 Accuracy . . . . .	8
2.2.2 Cost . . . . .	9
2.2.3 Robustness . . . . .	10
2.2.4 Scalability . . . . .	10
2.2.5 Summary . . . . .	10
 <b>3 Pairwise Clock Synchronization via Two-Way Message Exchange</b>	 <b>11</b>
3.1 Protocols . . . . .	11
3.1.1 One-Way Message Transmission . . . . .	11
3.1.2 Receiver-Receiver Message Exchange . . . . .	11
3.1.3 Two-Way Message Exchange . . . . .	12
3.2 Problem Formulation . . . . .	13
3.2.1 System Model . . . . .	13
3.2.2 Challenges . . . . .	14
3.3 Tools and Algorithms . . . . .	15

3.3.1	Estimation of Relative Offset . . . . .	15
3.3.2	Estimation of Relative Offset and Skew . . . . .	16
3.3.3	Estimation of Relative Offset, Skew, and Drift . . . . .	16
<b>4</b>	<b>Network-Wide Clock Synchronization via Distributed Schemes</b>	<b>17</b>
4.1	Problem Formulation for Distributed Clock Synchronization . . . . .	19
4.1.1	System Model . . . . .	19
4.1.2	Challenges . . . . .	20
4.2	Tools and Algorithms . . . . .	21
4.2.1	Algorithms Based on Matrix Theory . . . . .	21
4.2.2	Algorithms Based on Distributed Optimization . . . . .	21
<b>5</b>	<b>Conclusions and Future Work</b>	<b>23</b>
5.1	Contributions . . . . .	23
5.2	Conclusions . . . . .	24
5.3	Future Work . . . . .	24
5.4	Related Publications . . . . .	25
	<b>References</b>	<b>26</b>

## II Included papers 31

<b>A</b>	<b>On Clock Offset and Skew Estimation with Exponentially Distributed Delays</b>	<b>A1</b>
1	Introduction . . . . .	A2
2	System Model . . . . .	A3
2.1	Clock Value Relationship Model . . . . .	A3
2.2	Time Stamp Exchange Model . . . . .	A3
3	Proposed Clock Synchronization Algorithm . . . . .	A4
4	Simulation Results . . . . .	A8
5	Conclusion . . . . .	A12
	Appendix A . . . . .	A12
	References . . . . .	A12
<b>B</b>	<b>On Clock Offset and Skew Estimation with a Two-Way Message Exchange Mechanism</b>	<b>B1</b>
1	Introduction . . . . .	B2
2	System Model . . . . .	B4
3	Bounds and Algorithms for Known Delay Distributions . . . . .	B6
4	A Robust Synchronization Algorithm: L-rob . . . . .	B11
5	Simulation Results . . . . .	B13
6	Conclusion . . . . .	B17
	Appendix A . . . . .	B18

References . . . . .	B19
<b>C Random Broadcast Based Distributed Consensus Clock Synchronization for Mobile Networks</b>	<b>C1</b>
1 Introduction . . . . .	C2
2 System Model and Problem Formulation . . . . .	C4
3 The Proposed Synchronization Algorithm: RBDS . . . . .	C6
4 Simulation Results . . . . .	C14
5 Conclusion . . . . .	C17
Appendix A . . . . .	C17
References . . . . .	C23





# Part I

## Overview



# Chapter 1

## Introduction

### 1.1 Motivation

Time measurement has intrigued researchers for a long time. The first accurate clock started to appear in the 17th century after the scientific work pioneered by Galileo and expanded by Christian Huygens [1]. These scientists have helped to develop a better clock based on their theories of the motion of pendulums. More recent advances in the area of clocks include the development of the atomic clocks that are used in numerous applications including the Global Positioning System (GPS), and in the digital telephone communication network in order to provide more accurate time information [2]. Furthermore, it is also essential that different nodes in the network agree on the time their clocks are showing. Therefore, time agreement in the network is a critical issue, which allows for successful communications between nodes. The procedure to provide the common notion of time across all the nodes in the whole network is called *clock synchronization*. In order to achieve clock synchronization, it is necessary to establish links of communication between the clocks comprising the network. Those links can be wired or, more importantly for our research, wireless links.

The desire for ubiquitous communications, possibly when mobile, has led to increasing interests in wireless networks, where the clock synchronization has emerged as an indispensable component for many applications in the scenarios like sensor networks, ad hoc networks, vehicular networks and so on. Applications include (but not limited to)

- coordination of the actions across a distributed set of nodes in ad hoc mode;
- data fusion where data from each node is agglomerated to form a single meaningful result;
- time related transmission scheduling such as time division multiple access (TDMA);
- target tracking and localization where a common time scale should be used to measure distance.

While clock synchronization in wired networks is a well studied problem [3, 4], the wireless medium presents extra challenges. In wireless sensor networks (WSNs) which can contain several thousands of cheap and small sensors, the scalability and energy

conservation are two major considerations. Moreover, in wireless ad hoc networks which have no infrastructure or limited infrastructure support, self-organization is an important concern. Also, in vehicular networks, the high mobility will give rise to the requirement on robustness against node failures and dynamic topologies.

There has been extensive research on clock synchronization in wireless networks during the last few years. Several surveys [5–9] has been written about this issue, and a detailed explanation of the problems is given by Elson’s Ph.D. dissertation [10]. Various clock synchronization algorithms (also called clock synchronization protocols) have been proposed in the literature, which can be classified based on different criteria.

- *Pairwise vs network-wide synchronization*

- pairwise synchronization [9, 11] [Paper A] [Paper B]: the synchronization is designed for two nodes, i.e., synchronize one node to the other one, where the main principle is the accuracy and efficiency of parameter estimation;
- network-wide synchronization [12] [Paper C]: the synchronization is designed to synchronize multiple (more than 2) nodes in the network, where the main concern is the convergence speed and robustness against dynamic topology.

- *Pulse-based vs message-based synchronization*

- pulse-based synchronization [7, 13]: the synchronization is based on emitting the pulses of oscillators at the physical layer, which therefore needs additional information for the agreement on the time value;
- message-based synchronization [14, 15]: the synchronization is based on recording the local time of a node’s clock in a message (e.g., timestamp), which is then transmitted to other nodes via the wireless channel. Here the delays at the transmitter can introduce negative effects on the accuracy of synchronization.

- *Reference-based vs mutual synchronization*

- reference-based synchronization [14, 16, 17]: a group of clocks are synchronized to one (or more) reference clock, which can be implemented by both constructing hierarchical trees<sup>1</sup> and in distributed manners;
- mutual synchronization [18] [Paper C]: all the clocks align their time to one another without the need of reference clock, which is usually a distributed approach.

Additional classifications can be found in [5].

## 1.2 Scope and Aim of the Thesis

The scope of the thesis includes both pairwise and network-wide synchronization problems, with the assumption on message-based synchronization.

The aims of the thesis are to

---

<sup>1</sup>The reference clock is set as the root of the tree.

- establish clock models for short-term synchronization and long-term synchronization, respectively;
- select the performance metrics for clock synchronization;
- analyze the main challenges in pairwise synchronization, and propose clock synchronization algorithms based on parameter estimation techniques;
- discuss possible mathematical tools for network-wide synchronization, and propose distributed clock synchronization algorithms with consensus principles.

## 1.3 Thesis Outline

We start Chapter 2 of this thesis by introducing some general concepts and background on clock synchronization, which includes clock models and performance metrics. Then, in Chapter 3, the pairwise clock synchronization problem is studied. It is first formulated through the two-way message exchange mechanism, and then transformed into an estimation problem due to the randomness of transmission delays. Moreover, a set of synchronization algorithms are investigated with different considerations on clock models and delays' distributions. In Chapter 4, we study network-wide clock synchronization problem with distributed manners. We first formulate the problem for distributed clock synchronization, and then discuss some possible mathematical tools to solve this problem. Finally, the contributions of this thesis and several future directions are summarized in Chapter 5.



# Chapter 2

## General Concepts on Clock Synchronization

### 2.1 Clock Models

#### 2.1.1 Hardware Clock Models

According to the definition in [5], a hardware clock is an electronic device that counts oscillations at a particular frequency. Hence, a hardware clock usually consists of an oscillator and a counter. The oscillator is used to generate periodic events and the counter accumulates these events in order to obtain the measured time. For instance, the oscillator output can be modeled by a sinusoidal waveform,

$$S(t) = A(t) \sin \Phi(t), \quad (2.1)$$

where  $\Phi(t)$  is the phase,

$$A(t) = A + \Delta_A(t) \quad (2.2)$$

is the amplitude,  $\Delta_A(t)$  characterizes the amplitude variation, and  $t$  denotes the reference or global time. Note that the specific amplitude values of the oscillator are unimportant for our model. The instantaneous radian frequency function  $\dot{\Phi}(t)$  can be modeled in the form [19]

$$\dot{\Phi}(t) = \omega_0 + \sum_{k=0}^{M-1} \frac{L(k)}{k!} t^k + \dot{\xi}(t), \quad (2.3)$$

where  $\omega_0$  is a constant denoting the nominal value of the free running radian frequency of the oscillator.  $L(0)$  represents the initial radian frequency error (departure). This error arises from the uncertainty which exists in the initial setting of the oscillators. The  $L(k)$ 's ( $k = 1, \dots, M-1$ )<sup>1</sup> specify a set of time-independent values modeling the  $k$ th order radian

---

<sup>1</sup>  $M$  is a non-negative integer which indicates the order of frequency drifts we consider in the hardware clock model.

frequency drifts, and  $\dot{\xi}(t)$  is a stationary zero-mean random process characterizing the short-term oscillator instabilities.

The oscillator phase process can be obtained by integrating (2.3) from 0 to  $t$ . This results in

$$\Phi(t) = \Phi(0) + \omega_0 t + \sum_{k=1}^M \frac{L(k-1)}{k!} t^k + [\xi(t) - \xi(0)] \quad (2.4)$$

for  $M \geq 1$ . The 'value' of the hardware clock is obtained by dividing the oscillator phase by the nominal free-running radian frequency of the oscillator  $\omega_0$ . Correspondingly, the hardware clock value  $T(t)$  can be expressed as

$$T(t) \triangleq \frac{\Phi(t)}{\omega_0} = T(0) + t + q(1)t + \frac{q(2)}{2}t^2 + \sum_{k=3}^M \frac{q(k)}{k!}t^k + \Upsilon(t), \quad (2.5)$$

where  $T(0) = \Phi(0)/\omega_0$ ,  $q(k) = L(k-1)/\omega_0$  ( $k = 1, \dots, M$ ) are a set of values modeling the  $(k-1)$ th-order time drifts and  $\Upsilon(t) = [\xi(t) - \xi(0)]/\omega_0$  is, in general, a nonstationary stochastic process characterizing the short-term clock instabilities.

Even though (2.5) is a complete hardware clock model in the sense that it will be accurate for smooth phase functions as  $M$  is large and  $t$  is small, it is not necessary in practice. Firstly, the values of  $q(k)$  ( $k = 3, \dots, M$ ) are so small (compared to the clock resolution) [20] that it can be totally ignored. Moreover, in this thesis we assume  $\Upsilon(t) = 0$ . This is a reasonable assumption<sup>2</sup> since we are not dealing with the precise measurement of time, but rather the relative time synchronization of clocks in the network [20]. In this way, the important terms that characterize the performance of a hardware clock can be defined based on (2.5). These terms can also be called hardware *clock parameters*.

- *Offset*:  $\theta \triangleq T(0)$ .
- *Frequency*:  $f \triangleq 1 + q(1)$ .
- *Skew*:  $\rho \triangleq q(1)$ .
- *Drift*:  $D \triangleq q(2)/2$ .

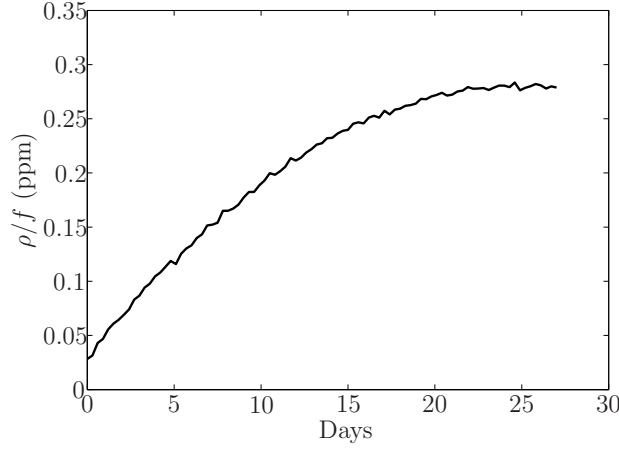
Note that offset, frequency, and drift are all determined by the hardware clock and cannot be measured or adjusted.

For a crystal oscillator commonly used in telecommunication radios, the reasonable values for the skew are [1, 100] part per million (ppm) relative to  $f$  [10]. Additionally, the drift is mainly caused by a gradual change in frequency over days or months, which is known as aging. An illustration of the aging effect is shown in Fig. 2.1<sup>3</sup>, which is inspired by [21]. Since the values of skew and drift usually have different scales, there should be different adopted hardware clock models in different applications.

<sup>2</sup>According to [21], for a typical quartz oscillator, the standard deviation of the short-term frequency variations is around  $10^{-12}$  by averaging 10 seconds.

<sup>3</sup>The aging rate of an oscillator is highest when it is first turned on [21].





**Figure 2.1:** An illustration of aging effects over time (days).

For long-term applications, i.e., the synchronized clocks need to be used for more than several hours, the effects of drift should be considered [22–24]. In this case, **clock model I** is presented as follows.

$$T(t) = Dt^2 + ft + \theta = Dt^2 + (1 + \rho)t + \theta. \quad (2.6)$$

For short-term applications, the effects of drift can be neglected. Then, there are two possible hardware clock models being utilized in the existing literature. Some works take both offset and skew into account [9, 11, 14–17, 25–36], which gives **clock model II** as

$$T(t) = ft + \theta = (1 + \rho)t + \theta. \quad (2.7)$$

Other works only consider offset [12, 15, 25], which introduces **clock model III**

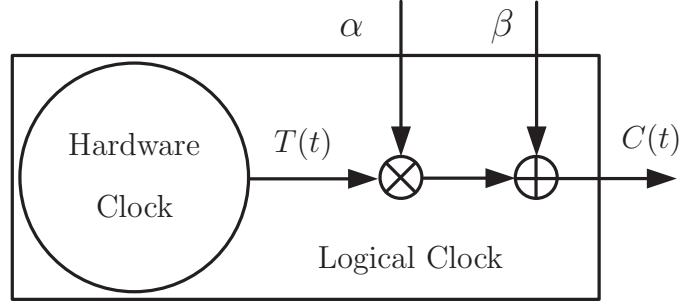
$$T(t) = t + \theta. \quad (2.8)$$

### 2.1.2 Logical Clock Models

Since the clock parameters of a hardware clock cannot be measured or adjusted manually, each node also maintains a logical clock whose value is a function of the current hardware clock value. In this thesis, we focus on the affine function, and calculate the logical clock value  $C(t)$  as

$$C(t) = \alpha T(t) + \beta, \quad (2.9)$$

where  $\alpha$  ( $\alpha > 0$ ) and  $\beta$  are control parameters updated by the synchronization algorithm. In this way, the logical clock value  $C(t)$  represents the synchronized time for each node. This relationship is also shown in Fig. 2.2.



**Figure 2.2:** The relationship between hardware clock and logical clock.

## 2.2 Performance Metrics

Even though a wide range of applications require clock synchronization, each application places different demands. For instance, some applications need high accuracy, e.g., T-DMA, while some applications have less requirement on the accuracy but more on the energy efficiency, e.g., power management in WSNs. To the best of our knowledge, there is no unique criterion to evaluate various clock synchronization schemes. In this section, we provide some possible metrics for the performance evaluation.

### 2.2.1 Accuracy

It is of great importance to synchronize clocks as accurately as possible. The accuracy, however, can be interpreted from different aspects.

#### Accuracy of Parameter Estimation

In pairwise clock synchronization, the objective is usually to synchronize one node to the other one by adjusting local clock parameters. In this way, the clock synchronization problem is inherently a parameter estimation problem. Even in network-wide synchronization, each node needs to update its clock parameters based on the timing messages received from other nodes. This is also related to parameter estimation if we consider that the received messages are inaccurate due to the delays and distortions.

In parameter estimation, a fairly common performance metric is the root mean square error (RMSE). The RMSE of an estimator  $\hat{z}$  is defined as

$$\text{RMSE} \triangleq \sqrt{\mathbb{E}[(\hat{z} - z)^2]}, \quad (2.10)$$

where  $z$  is the true value, and the expectation is taken over different realizations.

#### Accuracy of Synchronized Clock Value

What is finally being used is the synchronized clock value, and therefore its accuracy is another reasonable performance metric. Consider a network that consists of a set of nodes  $\mathcal{V} = \{1, 2, \dots, N\}$ . The synchronization error between a pair of nodes  $i$  and  $j$  is defined as

$$e_{ij}(t) \triangleq |C_i(t) - C_j(t)|, \quad i, j \in \mathcal{V}, \text{ and } i \neq j \quad (2.11)$$

where  $C_i(t)$  and  $C_j(t)$  are the synchronized clock values at global time  $t$  of nodes  $i$  and  $j$ , respectively. We model  $e_{ij}(t)$  as random process, which in general is nonstationary. The randomness comes from, e.g., randomness in transmissions of signals or timing messages (channel noise, packet errors, transmission delays, etc.), random network topologies, or random mobility.

Since  $|\mathcal{V}| = N$ , there are  $N(N-1)/2$  unique, nonzero synchronization error processes. These can be (partially) characterized by their marginal cumulative distribution functions (CDFs),

$$P_{e_{ij}(t)}(x) \triangleq \Pr \{e_{ij}(t) \leq x\}, \quad i, j \in \mathcal{V}, \quad i \neq j, \quad \text{and } t \in \mathbb{R}, \quad (2.12)$$

where the CDF is calculated over different pairs and different network topologies. From the CDF, we can calculate various metrics such as expected value, standard deviation, median or other percentiles.

To characterize a network, we can study the CDF of, e.g., the average error (over all unique pairs),

$$\bar{e}(t) \triangleq \frac{2}{N(N-1)} \sum_{i,j \in \mathcal{V}, j > i} e_{ij}(t), \quad (2.13)$$

or the worst-case error

$$e_{\max}(t) \triangleq \max_{i,j \in \mathcal{V}} e_{ij}(t). \quad (2.14)$$

That is, the CDFs are

$$P_{\bar{e}(t)}(x) \triangleq \Pr \{\bar{e}(t) \leq x\}, \quad (2.15)$$

$$P_{e_{\max}(t)}(x) \triangleq \Pr \{e_{\max}(t) \leq x\}, \quad (2.16)$$

where the CDFs are calculated over different network topologies, and we can compute expected values, moments, percentiles, etc., for these.

### 2.2.2 Cost

Another evaluation criterion, which is mainly a practical issue, is the cost of implementing an algorithm. In general, cost is commonly studied using the following metrics.

- Convergence time, which is the time needed by the algorithm to converge to a desired synchronization accuracy.
- Algorithm complexity, which determines the computational complexity of the algorithm in time and space.
- Power consumption, which is a combination of the power required to perform the local implementation of the synchronization algorithm, and the power consumed to send and receive timing messages.
- Communication overhead, which is the amount of transmitted information needed for the synchronization algorithm.

### 2.2.3 Robustness

As analyzed in Section 2.2.1, there are different sources for the randomness. Therefore, robustness is another important issue for clock synchronization algorithms. The considerations of robustness include

- robustness against random transmission delays with different levels and distributions;
- robustness against dynamic topologies with node failures and changing connections, which is considered for network-wide synchronization algorithms;
- robustness against packet losses regarding the timing messages.

### 2.2.4 Scalability

The network may consist of a few or many nodes. Hence, one way to assess the network-wide synchronization algorithms is to evaluate if they can perform well in both small networks and large networks, which is often referred as the scalability.

### 2.2.5 Summary

In conclusion, even though there is no unique way to compare different clock synchronization algorithms, some reasonable metrics, e.g., the metrics considered in this section, can be utilized for the performance evaluation.

## Chapter 3

# Pairwise Clock Synchronization via Two-Way Message Exchange

### 3.1 Protocols

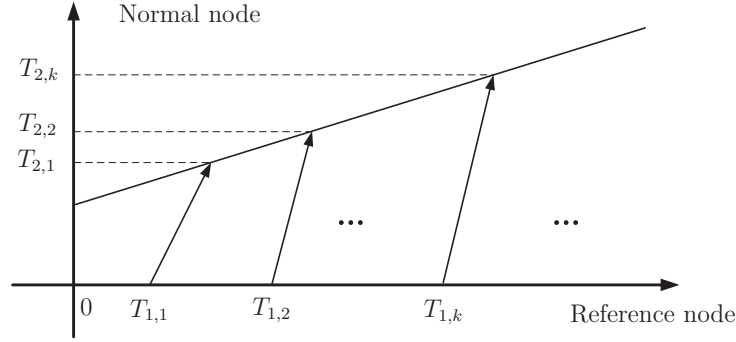
Clock synchronization can be achieved by transferring timing messages among different nodes, where the timing messages contain the timestamps recorded by the clock of the transmitter. There are three basic message transmission mechanisms: one-way message transmission, receiver-receiver exchange, and two-way message exchange.

#### 3.1.1 One-Way Message Transmission

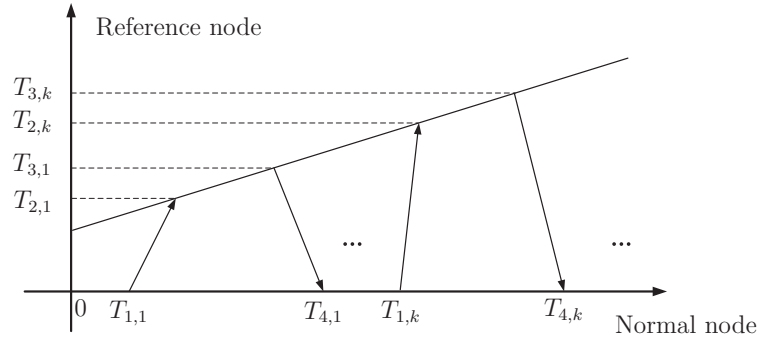
In one-way message transmission mechanism, the reference node records its current clock value and transmits it as a timestamp to the other node. Then the receiver (i.e., a normal node) measures its local clock value at the arrival of the timestamp [16]. This procedure is defined as a synchronization round (SR) in the one-way message transmission scheme. After some SRs, using the collected timestamps and local clock values, the normal node synchronizes itself to the reference node by estimating its relative clock parameters with regard to the reference node. The one-way message transmission mechanism is illustrated in Fig. 3.1, where  $\{T_{1,1}, T_{1,2}, \dots\}$  is provided by the reference node and  $\{T_{2,1}, T_{2,2}, \dots\}$  are the reception times recorded in the normal node. Note that  $T_{1,k}$  is in the reference node clock and  $T_{2,k}$  is in the normal node clock.

#### 3.1.2 Receiver-Receiver Message Exchange

The receiver-receiver message exchange mechanism [37] exploits the broadcast nature of the medium to synchronize the receivers with each other. A reference node broadcasts a message that arrives at roughly the same instant at different receivers; the time difference is mainly due to the difference between the propagation delays to the receivers, which may be neglected for some applications. The timestamp of the reception of the broadcast message is recorded at each receiver and exchanged with each other to calculate the relative clock parameters.



**Figure 3.1:** One-way message transmission.



**Figure 3.2:** Two-way message exchange.

### 3.1.3 Two-Way Message Exchange

The most widely used mechanism is the two-way message exchange [14], which identifies a two-way communication handshake protocol between two nodes sharing their timestamps in order to achieve synchronization. Here each two-way communication is defined as one SR. This mechanism is described in Fig. 3.2. In the  $k$ th SR, the normal node records its current clock value as timestamp  $T_{1,k}$  and sends a message to the reference node at the same time. The reference node records its clock value  $T_{2,k}$  at the reception of that message. Then the reference node measures its current clock value as  $T_{3,k}$  and sends another message containing  $T_{2,k}$  and  $T_{3,k}$ . Finally, the normal node records its clock value as  $T_{4,k}$  when receiving the message. Note that  $T_{1,i}$  and  $T_{4,i}$  are the timestamps provided by the normal node, while  $T_{2,i}$  and  $T_{3,i}$  are the timestamps recorded by the reference node.

In the following of this chapter, we will study the pairwise clock synchronization problem using the two-way message exchange mechanism.

## 3.2 Problem Formulation

### 3.2.1 System Model

Assume node  $i$ , which is the normal node, needs to synchronize to node  $j$ , which is the reference node. The clock values of the two nodes are

$$T_i = g_i(t) \quad \text{and} \quad T_j = g_j(t), \quad (3.1)$$

respectively. Here  $g_i(t)$  and  $g_j(t)$  are continuous, differentiable, and strictly increasing functions, since we assume clocks can neither stop nor run backwards. Moreover,  $g_i(t)$  and  $g_j(t)$  can be defined as different forms to represent the different clock models in Section 1. Specifically, if  $g_i(t)$  is a quadratic function with three parameters, it denotes the **clock model I**; if  $g_i(t)$  is an affine function with both slope and intercept, it indicates the **clock model II**; and if  $g_i(t)$  is an affine function with only intercept, it represents the **clock model III**. In this way, the clock value relationship between the two nodes can be expressed as

$$T_j = g_j(g_i^{-1}(T_i)) = g_{ji}(T_i), \quad (3.2)$$

where  $g_i^{-1}(t)$  is the inverse function of  $g_i(t)$ . We can expand the function  $g_{ji}(T_i)$  around the point  $T_i = T^{(0)}$  through a Taylor expansion

$$T_j = g_{ji}(T_i) \quad (3.3)$$

$$\begin{aligned} &= g_{ji}(T^{(0)}) + g'_{ji}(T^{(0)})(T_i - T^{(0)}) + \frac{g''_{ji}(T^{(0)})}{2}(T_i - T^{(0)})^2 \\ &\quad + \sum_{k=3}^{\infty} \frac{g_{ji}^{(k)}(T^{(0)})}{k!}(T_i - T^{(0)})^k, \end{aligned} \quad (3.4)$$

where  $T^{(0)}$  is usually set as node  $i$ 's clock value when it starts the synchronization, i.e.,  $T^{(0)} = T_{1,1}$  in Fig. 3.2. As analyzed in [19], the higher-order terms in the Taylor expansion in (3.4) are very small. Then, according to different applications and requirements, (3.4) can be approximated by the following three different models.

- *Relative offset and skew and drift model*

$$T_j \approx \bar{\theta} + (1 + \bar{\rho})(T_i - T^{(0)}) + \bar{D}(T_i - T^{(0)})^2, \quad (3.5)$$

where  $\bar{\theta} \triangleq g_{ji}(T^{(0)})$ ,  $\bar{\rho} \triangleq g'_{ji}(T^{(0)}) - 1$  and  $\bar{D} \triangleq g''_{ji}(T^{(0)})/2$  denote the relative offset, relative skew, and relative drift, respectively.

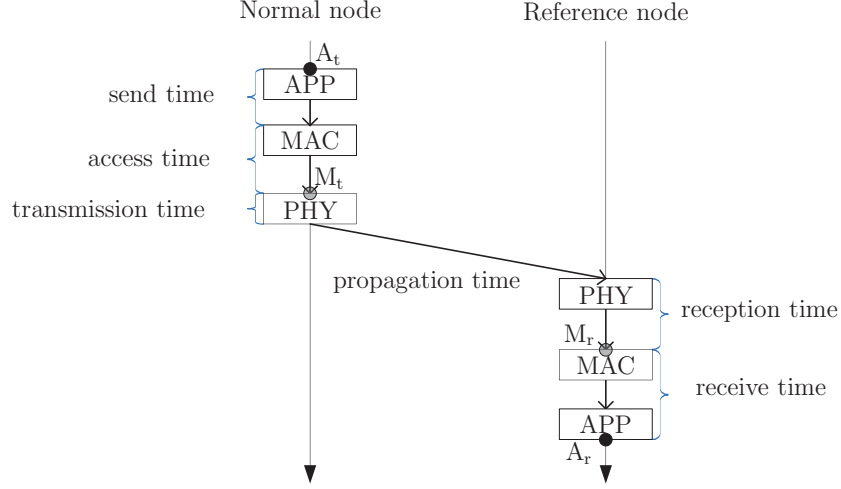
- *Relative offset and skew model*

$$T_j \approx \bar{\theta} + (1 + \bar{\rho})(T_i - T^{(0)}). \quad (3.6)$$

- *Relative offset model*

$$T_j \approx \bar{\theta} + T_i, \quad (3.7)$$

where  $\bar{\theta} \triangleq g_{ji}(T^{(0)}) - T^{(0)}$  denotes the relative offset.



**Figure 3.3:** Sources of transmission delays.

Note that (3.6) is a complete and accurate description of the relationship if both nodes have **clock model II**.

After modeling the clock value relationship, the objective of pairwise synchronization becomes the estimation of relative clock parameters in (3.5), (3.6) or (3.7), by utilizing the collected timestamps from the two-way message exchange mechanism.

### 3.2.2 Challenges

Even though we only consider two nodes now, clock synchronization is not an easy task. In the absence of the delay during the message transmission between two nodes, the receiver node will immediately know the relative clock value difference with respect to the transmitter node. Nonetheless, in practical wireless networks, various delays impose effects on the timestamp exchange procedure, which will complicate the clock synchronization.

Fig. 3.3 presents the possible sources of delays [14] [16], where each component is briefly analyzed as follows.

- **Send time:** the time spent at the sender to construct the message. Send time also accounts for the delay incurred by the packet to reach the medium access control (MAC) layer from the application (APP) layer. This delay is highly variable due to the software delays introduced by the underlying operating system.
- **Access time:** the delay incurred waiting for access to the transmit channel. This is perhaps the most critical factor contributing to the overall delay. Moreover, it is highly variable in nature and is specific to the MAC protocol employed by the sensor node.
- **Transmission time:** it refers to the time when a packet is transmitted bit by bit at the physical (PHY) layer. This delay is mainly deterministic.



- Propagation time: this is the actual time taken by the packet to traverse the wireless link from the sender to the receiver. It depends mainly on the distance between the two nodes.
- Reception time: this refers to the time taken in receiving the bits and passing them to the MAC layer. This is going to be mainly deterministic.
- Receive time: time to process the incoming message and to notify the receiver application. Its characteristics are similar to that of send time.

Depending on which point in Fig. 3.3 we obtain timestamps, the time stamping method can be divided into APP layer time stamping and MAC layer time stamping. In APP layer time stamping, the transmitter records its timestamp when the packet enters the APP layer (i.e., at the  $A_t$  point), then the receiver measures its timestamp just when the packet leaves the APP layer (i.e., at the  $A_r$  point). On the other hand, MAC layer time stamping means that the current timestamp is written into the message payload right before the first bit of the packet is sent to the PHY layer at the transmitter (i.e., at the  $M_t$  point), and the timestamp at the receiver side is recorded right after the first bit has arrived at the MAC layer (i.e., at the  $M_r$  point). Both time stamping methods have their pros and cons. The MAC layer time stamping can largely reduce the effects of transmission delays. However, it requires accessing the lower layers, which is not always possible when using standard or complex protocols [38].

The overall delays can be categorized into two classes: fixed delays and random delays. The characteristics of random delays depend on various applications and justifications, and thus no single model can fit for every case. Several probability density functions (PDFs) (e.g., Gaussian, exponential, Gamma, Weibull, and so forth) are utilized to model the random delays, where Gaussian and exponential are the two most widely used distributions. The Gaussian model is reasonable [37] if the delays are thought to be the addition of many independent random processes due to the central limit theorem, while, the exponential distribution is justified in [39].

## 3.3 Tools and Algorithms

Based on the two-way message exchange mechanism, many pairwise clock synchronization schemes have been designed over the past few years.

### 3.3.1 Estimation of Relative Offset

When only relative offset is considered, i.e., the relationship model in (3.7), [15] presents the maximum likelihood (ML) estimator under the assumption of Gaussian random delays, which coincides with the corresponding minimum variance unbiased (MVU) estimator. With exponentially distributed random delays, [40] states that the ML estimator is not unique when both fixed delays and the mean of random delays are known. On the other hand, [41] proposes the ML estimator when the fixed delays are not available, with the mean of random delays being either known or unknown; and [25] derives the MVU

estimator. Furthermore, the Weibull distributed random delays are considered in [42], and the associated uniformly MVU estimator is also proposed.

### 3.3.2 Estimation of Relative Offset and Skew

When both relative offset and skew exist, i.e., the relationship model in (3.6), the authors in [15] and [43] propose joint ML estimators for Gaussian random delays with known and unknown fixed delays, respectively. Also, the joint Cramér-Rao lower bound (CRLB) is derived in [15].

Assuming exponentially distributed random delays, [26] and [11] propose two different iterative ML estimators. Another ML estimator is derived in [44] by transforming the five dimensional optimization problem into a one dimensional problem. To further exploit the available range constraint on the relative skew, [Paper A] proposes a clock synchronization algorithm which directly uses MSE as the metric to be optimized. Moreover, since the regularity condition of deriving CRLB does not hold in the exponential case, [11] introduces an approximated CRLB, and [Paper B] deduces the Chapman-Robbins bound (CHRB).

More recently, without the assumption on the distribution of random delays, two robust synchronization algorithms are proposed in [45] and [Paper B], respectively.

### 3.3.3 Estimation of Relative Offset, Skew, and Drift

When the relative drift is also considered, i.e., the relationship model in (3.5), the authors in [22] derive the joint ML estimator for clock offset, skew and drift with exponentially distributed random delays, and we propose an algorithm in [23] to estimate these three parameters with arbitrary distribution of random delays.

## Chapter 4

# Network-Wide Clock Synchronization via Distributed Schemes

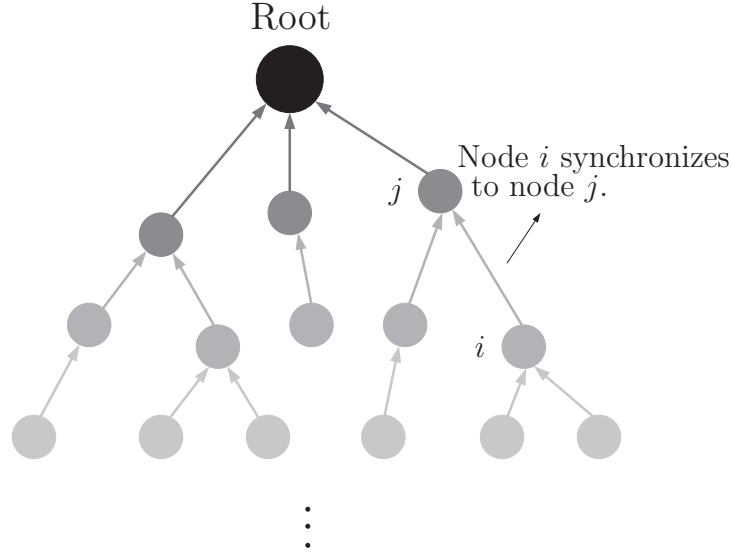
In contrast to Chapter 2, where the clock synchronization is investigated between a pair of nodes, this chapter considers the clock synchronization for multiple (more than 2) nodes, i.e., network-wide clock synchronization.

Some clock synchronization protocols [14, 16, 46] elect one (or more) reference node(s), and build a hierarchical tree with the reference node as the root. The synchronization is then carried out through the tree from the root to leaves, as shown in Fig. 4.1. To synchronize the nodes at adjacent levels, the pairwise clock synchronization is performed with two-way [14, 46] or one-way [16] message exchange mechanism. In these protocols, the estimation techniques discussed in Section 3.3 can be utilized directly for synchronizing two nodes at adjacent levels. Additionally, the receiver-receiver exchange mechanism for pairwise synchronization can also be extended to network-wide synchronization, which relies on the effective clustering of the nodes around the broadcast nodes [37].

The network-wide clock synchronization schemes described in the previous paragraph rely on the hierarchical structure, and is therefore sensitive to node failures and dynamic network topologies. For instance, if the cluster head leaves the network, a new one is required to be discovered and selected. Likewise, if the nodes are mobile, new hierarchies need to be constructed across the entire network for different time instances. These limitations will pose challenges on their applications in mobile or ad hoc networks.

An alternative approach to network-wide clock synchronization is based on distributed schemes, where nodes implement the same algorithm individually without relying on a network hierarchy. The distributed nature of these techniques can often result in improved scalability, and robustness to node failures as well as network mobility. There are mainly three design approaches.

- Converge-to-max, where a node only synchronizes to the transmitter which has greater clock value than the node's own local clock. Due to its simplicity, this protocol is the technique specified in IEEE 802.11 [12], which is called timing synchronization function (TSF). In 802.11 TSF, clock synchronization is achieved by periodical timing information exchange through beacon frames, which contain timestamps.



**Figure 4.1:** Network-wide clock synchronization via hierarchical tree.

The lack of scalability of TSF is first analyzed in [47]. To handle the scalability problem, the authors in [29, 47, 48] propose several methods to improve the TSF, where the basic principle is to make each node adjust its frequency of beacon transmission according to the received beacon messages. However, as addressed in [49], a common problem for all converge-to-max schemes is the contradiction between the *fastest node asynchronism* and the *time partitioning*. Since a node only synchronizes to a faster node, the clock value of the fastest node (a node with the greatest clock value in the network) will keep drifting away from other nodes, unless it becomes the beacon transmitter. This problem is called the fastest node asynchronism problem, which can be reduced by giving higher priority to the beacon transmissions of the node with a faster clock. Nevertheless, the different priority will result in the time partitioning problem, where the clock values in two groups of nodes can keep on drifting away from each other, even though they are connected.

- Reference-consensus in distributed manner [50, 51], where one or more reference nodes with accurate clocks exist and the goal is to disseminate the accurate clock value throughout the entire network in a distributed manner. In this mechanism, the reference node(s) will ignore the received timing messages, and the normal nodes will use the same algorithm to update its local logical clock based on the timing messages received from its neighbors.
- Arbitrary-consensus, where the goal is to synchronize all the clocks in the network to a common value. Note that it does not matter what this common value is. There are many work [30–34, 36, 52–56] [Paper C] in this direction. The main differences lie in the timing message transmission mechanism and the local clock adjustment algorithm. In the remaining of this chapter, we will focus on the arbitrary-consensus clock synchronization problem.

## 4.1 Problem Formulation for Distributed Clock Synchronization

### 4.1.1 System Model

We consider a network represented by a graph  $\mathcal{G}(\ell) = (\mathcal{V}, \mathcal{E}(\ell))$ , where the vertex set  $\mathcal{V} = \{1, 2, \dots, N\}$  contains  $N$  nodes, and the edge set  $\mathcal{E}(\ell)$  is defined as the set of available communication links during the  $\ell$ th SR. For network-wide clock synchronization, one SR is defined as the time interval between two intended broadcasts per node.

Each node in the network is equipped with a hardware clock, here we assume the **clock model II** in (2.7) is used. That is, for the  $i$ th node, we have the hardware clock value,

$$T_i(t) = f_i t + \theta_i, \quad \forall i \in \mathcal{V}, \quad (4.1)$$

where  $f_i$  indicates the hardware clock frequency, and  $\theta_i$  denotes the hardware clock offset. Besides, each node also maintains a logical clock value, i.e.,  $C_i(t)$  for the  $i$ th node. Based on (2.9), then,  $C_i(t)$  is calculated as:

$$C_i(t) = \alpha_i T_i(t) + \beta_i \quad (4.2)$$

$$= \alpha_i f_i t + \alpha_i \theta_i + \beta_i \quad (4.3)$$

$$= \hat{f}_i t + \hat{\theta}_i, \quad (4.4)$$

where  $\alpha_i$  ( $\alpha_i > 0$ ) and  $\beta_i$  are the  $i$ th node's control parameters updated by the synchronization algorithm (see Fig. 2.2), and

$$\hat{f}_i \triangleq \alpha_i f_i \quad (4.5)$$

$$\hat{\theta}_i \triangleq \alpha_i \theta_i + \beta_i \quad (4.6)$$

represent the logical clock frequency and logical clock offset.

The aim of network-wide clock synchronization is to attain an arbitrary-consensus between logical clocks through local interactions between nodes. Namely,

$$\lim_{t \rightarrow +\infty} \frac{C_i(t)}{C_v(t)} = 1, \quad \forall i \in \mathcal{V}, \quad (4.7)$$

where

$$C_v(t) = f_v t + \theta_v, \quad f_v > 0, \quad (4.8)$$

denotes the virtual consensus clock. For the  $i$ th node, the asymptotic consensus (4.7) is equivalent to concurrently achieving the following two consensus equations:

$$\lim_{t \rightarrow +\infty} \hat{f}_i = f_v, \quad (4.9)$$

$$\lim_{t \rightarrow +\infty} \hat{\theta}_i = \theta_v, \quad (4.10)$$

where the values of  $f_v$  and  $\theta_v$  are decided by  $\mathcal{E}(\ell)$ ,  $\{f_1, f_2, \dots, f_N\}$ , and  $\{\theta_1, \theta_2, \dots, \theta_N\}$  together.

### 4.1.2 Challenges

There are mainly three challenges regarding the distributed clock synchronization problem.

#### Required Timing Messages and Their Transmission Mechanisms

To utilize the broadcast nature of the wireless medium, in distributed clock synchronization, nodes broadcast timing messages which contain the timestamps recorded by the clock of the transmitter. These messages are in turn used to adjust the clocks of the receivers. When we are dealing with multihop networks, i.e., not every pair of two nodes can communicate with each other, the broadcasted messages can only reach the neighbors of the transmitter.

Some synchronization algorithms require the timing messages from all the neighbors, which can further be used in two different ways.

- Simultaneous update [17, 32, 33]: each node first collects the timing messages from all of its neighbors, and then adjusts its local logical clock by adopting these messages simultaneously.
- Sequential update [31, 34]: each node sequentially update its logical clock whenever it receives a timing message provided that it can receive the messages from all of its neighbors during each SR.

Both ways work well when we have a scheduled medium access protocol (e.g., TDMA), where the order of the timing message transmissions of different nodes is set. When we have a random access (e.g., carrier sense multiple access), however, these synchronization schemes are hard to implement.

When assuming random medium access protocol, a node can broadcast timing messages at any time in any order. Even though this assumption is more practical, it gives rise to new challenges, e.g., there is no information about how many and what neighbors a node can receive messages from during each SR. In this way, it will be more difficult to achieve the arbitrary-consensus mentioned above. Some work in this direction are reported in [17, 27–30].

#### Adjustment of Logical Frequency

Since the logical frequency basically represents the slope of the logical clock model (4.4), the following two requirements are sufficient for the appropriate update of the logical frequency: 1) the local node, e.g, the  $i$ th node, receives at least two timestamps from another node, e.g, the  $j$ th node; 2) the logical clocks of both the  $i$ th node and the  $j$ th node are not adjusted between the two receptions. Obviously the conditions can be satisfied if the  $i$ th node does not adjust its logical clock until it receives the second message from the  $j$ th node. However, this method will cause fairly slow convergence time, especially for dense networks.

## Transmission Delays

Even though most literature in network-wide clock synchronization utilize the MAC layer time stamping to largely reduce the effects of transmission delays, there could still be some delays remaining. When these remaining delays are also taken into account, how to guarantee the robustness of the synchronization algorithms, is another challenge.

## 4.2 Tools and Algorithms

Distributed network-wide clock synchronization is inherently a consensus problem [57] which has a long history in the field of distributed computing. As in most distributed algorithms, the main design issues are convergence speed and resilience to noise<sup>1</sup>. There are tools from different areas for solving consensus problems, for instance, matrix theory, optimization theory, and control theory. In this section, we will give a summary regarding the applications of tools (from matrix theory and optimization theory) in distributed clock synchronization.

### 4.2.1 Algorithms Based on Matrix Theory

As analyzed in Section 4.1.1, the consensus of the logical clocks is equivalent to the consensus of both logical offsets and logical frequencies. Assume  $\mathbf{z}$  a vector including the states of all nodes in the network. Here the state can be either the logical offset or the logical frequency. In this way, the update of  $\mathbf{z}$  can be written in a vector form

$$\mathbf{z}^{(\ell+1)} = \mathbf{P}^{(\ell)} \mathbf{z}^{(\ell)}, \quad (4.11)$$

where  $\ell$  is the discrete time index of the SR,  $\mathbf{P}^{(\ell)}$  is the transfer matrix which depends on the current edge set  $\mathcal{E}(\ell)$  and local operations per node. The formulation (4.11) assumes that the new state updates are linear combinations of the previous states. Therefore, the key issue in clock synchronization is to derive the matrix  $\mathbf{P}^{(\ell)}$  by designing the local operations per node. Moreover, the authors in [58] and [59] present the sufficient conditions for  $\mathbf{P}^{(\ell)}$  to achieve the consensus in undirected graphs and directed graphs, respectively. Note that the derivation of the conditions does not take the noise into account. Following the conditions in [58] and [59], [30–34, 36, 52] and [Paper C] propose different clock synchronization algorithms by different local operations.

### 4.2.2 Algorithms Based on Distributed Optimization

As shown in [60], the consensus problem can be formulated as an optimization problem, where the optimal solution is the average consensus. The authors in [61] apply the alternating direction multiplier method (ADMM) into the distributed clock synchronization problem, where ADMM is an iterative algorithm for solving convex minimization problems. Furthermore, in order to reduce the effects of noise, [53–56] formulate optimization

---

<sup>1</sup>There are different sources for the noise, such as measurement noise, packet errors, and transmission delay, where the delay is the key factor in clock synchronization problems.

problems with noisy timing messages, where the solution can be used to optimize the update of logical clock parameters.



# Chapter 5

## Conclusions and Future Work

This thesis studies clock synchronization problem for wireless networks, where both pairwise clock synchronization and network-wide clock synchronization are investigated. In pairwise clock synchronization, we first analyze the challenges and transform the synchronization problem into an estimation problem. Then we utilize estimation techniques to propose several synchronization algorithms and performance bounds. In network-wide clock synchronization, we first formulate the problem for distributed clock synchronization, and then discuss some possible mathematical tools to solve this problem. Moreover, we also propose several network-wide synchronization algorithms based on consensus techniques.

### 5.1 Contributions

The main contributions of this thesis can be found in the three appended papers.

1. **Paper A: “On clock offset and skew estimation with exponentially distributed delays”**

In this paper, we investigate the pairwise clock synchronization problem based on a two-way timing message exchange mechanism with exponentially distributed random delays. Without knowing the fixed delays, a novel synchronization scheme is proposed for the linear clock model, which works well in both symmetric and asymmetric links. In the proposed algorithm, the clock skew and offset are estimated by directly utilizing mean square error as the metric to be optimized. This consideration results in significant performance improvements compared to the existing synchronization methods, especially for a small number of observations and large standard deviation of the random delays.

2. **Paper B: “On clock offset and skew estimation with a two-way message exchange mechanism”**

This paper studies the pairwise clock synchronization problem based on a two-way timing message exchange mechanism with unknown fixed delays. Different distributions of random delays are considered. For a Gaussian distribution, we derive the Cramér-Rao lower bound for the joint estimation of clock offset and skew, and propose a linear-biased estimator to improve the precision. For an exponential distribution, we extend

our previous work by deriving the Chapman-Robbins bound. Moreover, a simple and robust estimation algorithm is also proposed to handle arbitrary delay distributions, which can achieve a good tradeoff between accuracy and complexity. Finally, we present simulation results to evaluate different clock synchronization methods. It is shown that the proposed synchronization algorithms exhibit superiority over the existing schemes in most scenarios, especially for small number of observations or large delay variances.

### 3. Paper C: “Random broadcast based distributed consensus clock synchronization for mobile networks”

Clock synchronization is a crucial issue for mobile ad hoc networks due to the dynamic and distributed nature of these networks. In this paper, employing affine models for local clocks, we proposed a random broadcast based distributed consensus clock synchronization algorithm. In the absence of transmission delays, we theoretically prove the convergence of the proposed scheme, which is further illustrated by the numerical results. On the other hand, when the transmission delays are also taken into account, the proposed approach still shows resilience. Besides, it is also concluded from the simulations that the proposed scheme scales well with the number of nodes, and is robust to transmission delays as well as different accuracy requirements.

## 5.2 Conclusions

To conclude, in pairwise clock synchronization, the problem can be formulated as an estimation problem. In this scenario, biased estimators, which can be derived based on the knowledge of the range constraint on the clock skew, are useful to improve the RMSE of the estimations on clock offset and skew. However, since the biased estimators utilize the minimax idea, their performances are not straightforward to interpret. For example, the RMSE of the biased estimator is not a monotonically decreasing function as the number of observations increases or the delay variance decreases.

When it comes to network-wide clock synchronization, more considerations need to be taken into account, such as the ability of achieving consensus, convergence time, computational complexity, communication overhead, scalability to the number of nodes, and robustness against transmission delays. Therefore, no single algorithm can beat others from all perspectives. Nevertheless, if we focus on the ability to achieve consensus when there is no delay, and the robustness against delays when they are also considered, [Paper C] proposes a network-wide synchronization algorithm which performs well in both scenarios with and without delay.

## 5.3 Future Work

In the future, we will continue our study on network-wide clock synchronization. Ongoing research and some possible future directions are as follows.

- In Paper C, we assume there is no reference clock and our goal is to synchronize all the clocks to an arbitrary common value. In some scenarios, on the other hand, one

or several reference clocks do exist and the normal clocks are required to synchronize to the reference clock(s). In this case, what is the theoretical relationship between the convergence speed and the number of reference clocks? Furthermore, how to design distributed synchronization algorithms to speed up the convergence as much as possible?

- In pairwise clock synchronization, i.e., Paper A and Paper B, our focus is to reduce the effects of random transmission delays. While in network-wide clock synchronization, i.e., Paper C, we mainly study the convergence under the scenario without delay and then evaluate algorithms' robustness to delays through simulation. Nevertheless, if we take into account the statistics of random delays for designing distributed synchronization algorithm, the synchronization accuracy can be further improved. This can be somehow considered as the combination of parameter estimation and consensus techniques.

## 5.4 Related Publications

Other related publications by the author, which are not included in this thesis, are listed below.

- [C1] W. Sun, E. G. Ström, F. Brännström, and D. Sen, "Long-term clock synchronization in wireless sensor networks with arbitrary delay distributions," in *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, Anaheim, California, USA, Dec. 2012.
- [C2] W. Sun, M. R. Gholami, E. G. Ström, and F. Brännström, "Distributed clock synchronization with application of D2D Communication without infrastructure," in *Proceedings of IEEE Global Communications Conference (GLOBECOM) Workshop*, Atlanta, GA, USA, Dec. 2013.
- [C3] Y. Sui, Z. Ren, W. Sun, T. Svensson, and P. Fertl, "Performance study of fixed and moving relays for vehicular users with multi-cell handover under co-channel interference," in *Proceedings of IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, Las Vegas, Nevada, USA, Dec. 2013.

## References

- [1] D. M. Considine, *Van Nostrand's scientific encyclopedia*. New York: McGraw-Hill, 1995.
- [2] P. Kartaschoff, "Synchronization in digital communications networks," *Proceedings of the IEEE*, vol. 79, no. 7, pp. 1019–1028, July 1991.
- [3] D. L. Mills, "Network time protocol, specification, implementation and analysis," University of Delaware, Tech. Rep., March 1992.
- [4] *Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE Std.1588 Std., March 2008.
- [5] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: A survey," University of Illinois at Chicago, Tech. Rep., March 2005.
- [6] B. Sadler and A. Swami, "Synchronization in sensor networks: an overview," in *Proc. IEEE Military Communications Conf.*, October 2006.
- [7] O. Simeone, U. Spagnolini, Y. B. Ness, and S. Strogatz, "Distributed synchronization in wireless networks," *IEEE Signal Processing Magazine*, vol. 25, no. 5, pp. 81–97, September 2008.
- [8] I. K. Rhee, J. Lee, J. Kim, E. Serpedin, and Y. C. Wu, "Clock synchronization in wireless sensor networks: An overview," *Sensors*, vol. 9, no. 1, pp. 56–85, January 2009.
- [9] Y. Wu, Q. M. Chaudhari, and E. Serpedin, "Clock synchronization of wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 124–138, January 2011.
- [10] J. E. Elson, "Time synchronization in wireless sensor networks," Ph.D. dissertation, UCLA, 2003.
- [11] M. Leng and Y. Wu, "On joint synchronization of clock offset and skew for wireless sensor networks under exponential delay," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, June 2010, pp. 461–464.
- [12] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification*, IEEE Std 802.11 Std., 1999.
- [13] G. W. Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, "Firefly-inspired sensor network synchronicity with realistic radio effects," in *Proc. International Conf. on Embedded Networked Sensor Systems*, November 2005, pp. 142–153.
- [14] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proc. International Conf. on Embedded Networked Sensor Systems*, November 2003, pp. 138–149.

- [15] K. Noh, Q. M. Chaudhari, E. Serpedin, and B. W. Suter, "Novel clock phase offset and skew estimation using two-way timing message exchanges for wireless sensor networks," *IEEE Trans. Comm.*, vol. 55, no. 4, pp. 766–777, April 2007.
- [16] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The flooding time synchronization protocol," in *Proc. International Conf. Embedded Networked Sensor Systems*, November 2004, pp. 39–49.
- [17] R. Solis, V. S. Borkar, and P. R. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," in *Proc. IEEE Decision and Control Conf.*, December 2006, pp. 2734–2739.
- [18] C. H. Rentel and T. Kunz, "A mutual network synchronization method for wireless ad hoc and sensor networks," *IEEE Trans. on Mobile Computing*, vol. 7, no. 5, pp. 633–646, May 2008.
- [19] W. C. Lindsey, F. Ghazvinian, W. C. Hagmann, and K. Dessouky, "Network synchronization," *Proceedings of IEEE*, vol. 73, no. 10, pp. 1445–1467, October 1985.
- [20] C. H. Rentel, "Network time synchronization and code-based scheduling for wireless ad hoc networks," Ph.D. dissertation, Carleton University, January 2006.
- [21] [Online]. Available: <http://www.leapsecond.com/pdf/an200-2.pdf>
- [22] Q. Chaudhari and E. Serpedin, "Clock estimation for long-term synchronization in wireless sensor networks with exponential delays," *EURASIP Journal on Advances in Signal Processing*, January 2008.
- [23] W. Sun, E. G. Ström, F. Brännström, and D. Sen, "Long-term clock synchronization in wireless sensor networks with arbitrary delay distributions," in *Proc. IEEE Global Communications Conf. (GLOBECOM)*, December 2012.
- [24] Y. Quan and G. Liu, "Drifting clock model for network simulation in time synchronization," in *Proc. IEEE International Conf. on Innovative Computing Information and Control*, June 2008, pp. 385–388.
- [25] Q. M. Chaudhari, E. Serpedin, and K. Qaraqe, "On minimum variance unbiased estimation of clock offset in a two-way message exchange mechanism," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2893–2904, June 2010.
- [26] Q. Chaudhari, E. Serpedin, and K. Qaraqe, "On maximum likelihood estimation of clock offset and skew in networks with exponential delays," *IEEE Trans. Signal Processing*, vol. 56, no. 4, pp. 1685–1697, April 2008.
- [27] D. Zhou and T. H. Lai, "A scalable and adaptive clock synchronization protocol for IEEE 802.11-based multihop ad hoc networks," in *Proc. IEEE Mobile Adhoc and Sensor Conf.*, November 2005, pp. 558–565.

- [28] —, “An accurate and scalable clock synchronization protocol for IEEE 802.11-based multihop ad hoc networks,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 18, no. 12, pp. 1797–1808, December 2007.
- [29] H. K. Pande, S. Thapliyal, and L. C. Mangal, “A new clock synchronization algorithm for multi-hop wireless ad hoc networks,” in *Proc. IEEE International Conf. on Distributed Computing Systems*, December 2010.
- [30] L. Schenato and F. Fiorentin, “Average timesynch: A consensus-based protocol for clock synchronization in wireless sensor networks,” *Automatica*, vol. 47, no. 9, pp. 1878–1886, September 2011.
- [31] M. K. Maggs and S. G. Okeefe, “Consensus clock synchronization for wireless sensor networks,” *IEEE sensors Journal*, vol. 12, pp. 2269–2277, June 2012.
- [32] P. Sommer and R. Wattenhofer, “Gradient clock synchronization in wireless sensor networks,” in *Proc. IEEE International Conf. on Information Processing in Sensor Networks*, April 2009, pp. 17–48.
- [33] Q. Li and D. L. Daniela, “Global clock synchronization in sensor networks,” *IEEE Trans. on Computers*, vol. 55, no. 2, pp. 214–226, February 2006.
- [34] A. C. Pinho, D. R. Figueiredo, and F. M. G. Franga, “A robust gradient clock synchronization algorithm for wireless sensor networks,” in *Proc. IEEE International Conf. on Communication Systems and Networks*, January 2012.
- [35] W. Su and I. F. Akyildiz, “Time-diffusion synchronization protocol for wireless sensor networks,” *IEEE/ACM Trans. on Networking*, vol. 13, no. 2, pp. 384–397, April 2005.
- [36] W. Sun, M. R. Gholami, E. G. Ström, and F. Brännström, “Distributed clock synchronization with application of D2D communication without infrastructure,” in *Proc. IEEE Global Communications Conf. (GLOBECOM) Workshop*, December 2013.
- [37] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” in *Proc. 5th Symp. Operat. Syst. Design Implement.*, December 2002, pp. 147–163.
- [38] A. Marco, R. Casas, J. L. Ramos, V. Coarasa, A. Asensio, and M. S. Obaidat, “Synchronization of multihop wireless sensor networks at the application layer,” *IEEE Wireless Comm.*, vol. 18, no. 1, pp. 82–88, February 2011.
- [39] C. Bovy, H. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, and P. Miegheem, “Analysis of end-to-end delay measurements in internet,” in *Proc. Passive and Active Measurements Workshop*, March 2002, pp. 26–33.
- [40] H. S. Abdel-Ghaffar, “Analysis of synchronization algorithms with time-out control over networks with exponentially symmetric delays,” *IEEE Trans. on Comm.*, vol. 50, no. 10, pp. 1652–1661, October 2002.



- [41] D. Jeske, “On maximum-likelihood estimation of clock offset,” *IEEE Trans. on Communications*, vol. 53, no. 1, pp. 53–54, January 2005.
- [42] A. Ahmad, A. Noor, E. Serpedin, H. Nounou, and M. Nounou, “On clock offset estimation in wireless sensor networks with weibull distributed network delays,” in *Proc. IEEE International Conf. on Pattern Recognition (ICPR)*, August 2010, pp. 2322–2325.
- [43] M. Leng and C. Wu, “On clock synchronization algorithms for wireless sensor networks under unknown delay,” *IEEE Trans. on Vehicular Technology*, vol. 59, no. 1, January 2010.
- [44] J. Li and D. R. Jeske, “Maximum likelihood estimators of clock offset and skew under exponential delays,” *Special issue on the 6th International Symposium on Business and Industrial Statistics*, vol. 25, no. 4, pp. 445–459, July 2009.
- [45] Q. M. Chaudhari, “A simple and robust clock synchronization scheme,” *IEEE Trans. on Comm.*, vol. 60, no. 2, pp. 328–332, February 2012.
- [46] J. V. Greunen and J. Rabaey, “Lightweight time synchronization for sensor networks,” in *Proc. ACM International Conf. Wireless Sensor Networks and Applications*, September 2003, pp. 11–19.
- [47] L. Huang and T. H. Lai, “On the scalability of IEEE 802.11 ad hoc networks,” in *Proc. ACM International Symp. on Mobile Ad Hoc Networking & Computing*, June 2002, pp. 173–182.
- [48] J. P. Sheu, C. M. Chao, and C. W. Sun, “A clock synchronization algorithm for multi-hop wireless ad hoc networks,” in *Proc. IEEE International Conf. on Distributed Computing Systems*, March 2004, pp. 574–581.
- [49] J. So and N. Vaidya, “MTSF: A timing synchronization protocol to support synchronous operations in multihop wireless networks,” University of Illinois at Urbana-Champaign, Tech. Rep., January 2004.
- [50] R. Fan, I. Chakraborty, and N. Lynch, “Clock synchronization for wireless networks,” in *Proc. International Conf. on Principles of Distributed Systems*, December 2004, pp. 400–414.
- [51] K. Iwanicki, M. V. Steen, and S. Voulgaris, “Gossip-based clock synchronization for large decentralized systems,” in *Proc. IEEE International Workshop on Self-Managed Networks, Systems and Services*, June 2006, pp. 28–42.
- [52] B. J. Choi, H. Liang, X. Shen, and W. Zhuang, “DCS: Distributed asynchronous clock synchronization in delay tolerant networks,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 23, no. 3, pp. 1045–1059, March 2012.
- [53] N. M. Freris and A. Zouzias, “Fast distributed smoothing for network clock synchronization,” in *Proc. IEEE Conf. on Decision and Control*, 2012.

- [54] I. Shames and A. N. Bishop, “Distributed relative clock synchronization for wireless sensor networks,” in *Preprints of the 18th IFAC World Congress*, August 2011.
- [55] O. Gurewitz, I. Cidon, and M. Sidi, “Network classless time protocol based on clock offset optimization,” *IEEE/ACM Trans. on Networking*, vol. 14, no. 4, pp. 1063–6692, August 2006.
- [56] A. Giridhar and P. Kumar, “Distributed clock synchronization over wireless networks: Algorithms and analysis,” in *Proc. IEEE Conf. on Decision and Control*, December 2006, pp. 4915–4920.
- [57] R. O. Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, January 2007.
- [58] A. Jadbabaie, L. Jie, and A. S. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” *IEEE Trans. on Automatic Control*, vol. 48, no. 6, pp. 988–1001, June 2003.
- [59] W. Ren and R. Beard, “Consensus seeking in multiagent systems under dynamically changing interaction topologies,” *IEEE Trans. on Automatic Control*, vol. 50, no. 5, pp. 655–661, May 2005.
- [60] T. Erseghe, D. Zennaro, E. Dall’Anese, and L. Vangelista, “Fast consensus by the alternating direction multipliers method,” *IEEE Trans. on Signal Processing*, vol. 59, no. 11, pp. 5523–5537, October 2011.
- [61] D. Zennaro, E. Dall’Anese, T. Erseghe, and L. Vangelista, “Fast clock synchronization in wireless sensor networks via ADMM-based consensus,” in *Proc. IEEE International Symp. on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, May 2011, pp. 148–153.